PROG20799 Data Structures Exercise

Complete the following program. Please follow the C99 language standard, and use proper coding style and standards as you would for an assignment. When done submit your **.c** file(s) using the appropriate SLATE dropbox.

Bracket Checker: A C module called charStack.c (and header file charStack.h) has been provided, please download these files from SLATE. The stack module implements a stack data structure which holds characters (similar to the stack we studied in class except it holds char instead of int). Write another module called checker.c with a main function that does the following.

- Input a string from the user.
- Check if the string contains balanced round/square/curly brackets ( ) [ ] { }. This means that every opening bracket has a matching closing bracket of the correct type.
- If the brackets are balanced print "Brackets are correct", otherwise print "Error".

You can use the following algorithm:

a) Loop through the characters in the string.
b) If you encounter an opening bracket '(' or '[' or '{' then **push** it on the stack.
c) If you encounter a closing bracket ')' or ']' or '}' then **pop** a bracket off the stack. If the bracket popped is the matching type of *opening* bracket then brackets are balanced, otherwise there's an error.
d) When the loop finishes ensure that the stack is empty, otherwise there's an error.

Note that you could even use this program to check for correct use of brackets in C or Java source code! Write additional functions in checker.c if you require. Do not make any changes in charStack.c or charStack.h.